



**University of
Zurich**^{UZH}

**Zurich Open Repository and
Archive**

University of Zurich
University Library
Strickhofstrasse 39
CH-8057 Zurich
www.zora.uzh.ch

Year: 2008

Modellierung in der Lehre an Hochschulen: Thesen und Erfahrungen

Glinz, M

Abstract: Nicht erst seit Modellierung zum Modewort geworden ist, ist Modellierung ein wichtiges Thema in der Informatikausbildung. Dieser Beitrag beleuchtet in zwölf Thesen das Warum, Was, Wie viel, Wo, Wann und Wie von Modellierung in der Lehre im Rahmen informatikbezogener Studiengänge. Die Thesen basieren auf der Erfahrung des Verfassers mit einer seit über zehn Jahren gehaltenen Modellierungsvorlesung an der Universität Zürich sowie dem Gedankenaustausch mit anderen Hochschullehrerinnen und -lehrern im Rahmen der GI-Workshopreihe „Modellierung“. Zusätzlich liefert eine Befragung von Absolventinnen und Absolventen an der Universität Zürich empirische Evidenz für eine Reihe der Thesen.

DOI: <https://doi.org/10.1007/s00287-008-0273-x>

Posted at the Zurich Open Repository and Archive, University of Zurich

ZORA URL: <https://doi.org/10.5167/uzh-7368>

Journal Article

Published Version

Originally published at:

Glinz, M (2008). Modellierung in der Lehre an Hochschulen: Thesen und Erfahrungen. Informatik-Spektrum, 31(5):425-434.

DOI: <https://doi.org/10.1007/s00287-008-0273-x>

Modellierung in der Lehre an Hochschulen: Thesen und Erfahrungen

Martin Glinz

Entsprechend dem gestiegenen Stellenwert der Modellierung in der Informatik und ihren Anwendungen ist Modellierung heute ein wichtiges Thema in jedem informatikbezogenen Studiengang; das heißt sowohl in den klassischen Informatik-Curricula wie auch in Studiengängen der Angewandten Informatik, der Softwaretechnik oder der „Bindestrich“-Informatik, zu deren prominentesten Vertretern die Wirtschaftsinformatik gehört. Wenn nun im Folgenden vereinfachend von „Informatikstudium“ die Rede ist, so ist dies umfassend im Sinn der genannten Studiengänge gemeint.

Dies spiegelt sich auch in den einschlägigen Empfehlungen für Informatik-Curricula. Beispielsweise ist Modellierung im ACM/IEEE Computer Science Curriculum eine der sechs von Bachelorabsolventen erwarteten kognitiven Fähigkeiten [2, S. 64]. In den GI-Empfehlungen für Bachelor- und Masterprogramme [9, S. 8] ebenso wie in den Empfehlungen des Fakultätentags Informatik [8, S. 9] gehören das Denken in Modellen und die Fähigkeit zur Modellbildung zu den Ausbildungszielen.

Dieser Beitrag beschäftigt sich im Sinn eines Thesenpapiers mit den sechs W-Fragen der Modellierung in der Lehre: Warum?,

Gedankenaustausch mit anderen Hochschullehrerinnen und -lehrern, insbesondere im Rahmen der Workshopreihe „Modellierung“ und Software Engineering im Unterricht der Hochschulen (SEUH) sowie einer Befragung von Absolventinnen und Absolventen des Studiengangs Informatik Richtung Wirtschaftsinformatik an der Universität Zürich.

Zunächst gilt es, den hier verwendeten Modellbegriff zu präzisieren und abzugrenzen. Modellierung ist ein umfassendes Phänomen, das untrennbar mit menschlicher Erkenntnis an sich verbunden ist: Unsere Erkenntnisse und unsere Orientierung in dieser Welt sind letztlich das fortlaufende Bilden und Interpretieren von Modellen. Daraus zu schließen, Modellierung sei im Rahmen der Philosophie zu betrachten und zu lehren, ist jedoch verfehlt und wird der zentralen Rolle, welche Modellierung in der Informatik spielt, nicht gerecht.

Als Basis für den Umgang mit Modellen in der Informatik hat sich der Modellbegriff von Stachowiak [15] bewährt; er wird auch in diesem Artikel verwendet. Für eine Reflexion über den Modellbegriff sei auf Ludewig [13] verwiesen.

Definition. Ein Modell ist ein konkretes oder gedankliches Abbild von etwas oder ein konkretes oder gedankliches Vorbild für etwas. Dabei sind drei Merkmale konstituierend: 1. das Abbildungsmerkmal, d.h. es gibt ein Gebilde (Original genannt),

Was?, Wie viel?, Wo?, Wann? und Wie?. Betrachtet wird primär die Situation an Universitäten und technischen Hochschulen. Die Aussagen dieses Artikels stützen sich im Wesentlichen auf drei Quellen: Der Erfahrung des Autors mit einer eigenen Modellierungsvorlesung seit 1996 [11, 12], dem

DOI 10.1007/s00287-008-0273-x
© Springer-Verlag 2008

Martin Glinz
Universität Zürich, Institut für Informatik,
Binzmühlestrasse 14, 8050 Zürich, Schweiz
E-Mail: glinz@ifi.uzh.ch

Zusammenfassung

Nicht erst seit Modellierung zum Modewort geworden ist, ist Modellierung ein wichtiges Thema in der Informatikausbildung. Dieser Beitrag beleuchtet in zwölf Thesen das Warum, Was, Wie viel, Wo, Wann und Wie von Modellierung in der Lehre im Rahmen informatikbezogener Studiengänge. Die Thesen basieren auf der Erfahrung des Verfassers mit einer seit über zehn Jahren gehaltenen Modellierungsvorlesung an der Universität Zürich sowie dem Gedankenaustausch mit anderen Hochschullehrerinnen und -lehrern im Rahmen der GI-Workshopreihe „Modellierung“. Zusätzlich liefert eine Befragung von Absolventinnen und Absolventen an der Universität Zürich empirische Evidenz für eine Reihe der Thesen.

dessen Ab- oder Vorbild das Modell ist; 2. das Verkürzungsmerkmal, d.h. nur die im Kontext der Modellbildung relevanten Merkmale des Originals werden auf das Modell abgebildet; 3. das pragmatische Merkmal, d.h. das Modell wird in einem spezifischen Kontext für einen spezifischen Verwendungszweck geschaffen.

Bemerkung. Oft wird Abstraktion als konstituierendes Merkmal von Modellen betrachtet. Dies wird in obiger Definition durch das Verkürzungsmerkmal und das pragmatische Merkmal ausgedrückt. Ein Modell als Abstraktion eines Originals zu definieren, greift jedoch zu kurz: Zeichen beispielsweise sind Abstraktionen, aber keine Modelle.

Abgrenzung 1. Der Modellbegriff der mathematischen Logik, wo eine Interpretation einer Menge von Axiomen als Modell dieses Axiomensystems bezeichnet wird, fällt nicht unter den hier betrachteten Modellbegriff.

Abgrenzung 2. In der Informatik finden wir Modelle einerseits als Informatikartefakte bzw. als Mittel zur Erstellung oder zum Verstehen von Informatikartefakten und andererseits als Mittel zur Beschreibung und Berechnung von Systemen oder Phänomenen eines Anwendungsbereichs, zum Beispiel ein Wettermodell, ein Modell der Steifheit einer Baukonstruktion oder ein Modell für

Lastverteilung und Wartezeiten in einem verteilten Verarbeitungssystem.

Da die letztere Art von Modellen primär im Fokus eines Anwendungsbereichs liegt und spezieller mathematischer Methoden bedarf (vor allem Differenzialgleichungen und stochastische Methoden), sollte die zugehörige Modellierungslehre eher den jeweiligen Anwendungsbereichen oder der angewandten Mathematik zugeordnet werden als der Informatik. Dieser Artikel fokussiert daher auf Modelle als Informatikartefakte oder als Mittel zur Erstellung oder zum Verstehen von Informatikartefakten.

Die „sechs W“ der Modellierung in der Lehre

Warum Modellierung in der Lehre?

Informatik ist eine Disziplin, die sich mit dem Verstehen und Lösen von komplexen Problemen beschäftigt und dabei weitestgehend immaterielle Lösungsmittel verwendet. Da Modellierung ein entscheidendes Hilfsmittel für den Umgang mit Komplexität ist, gehört die Fähigkeit zur Bildung von Modellen und zum Umgang mit Modellen zu den unabdingbaren kognitiven Fähigkeiten von Informatikerinnen und Informatikern [2, 8, 9]. Hinzu kommt, dass Modellierung ein Kernbestandteil zentraler Gebiete der Informatik ist, insbesondere der Softwaretechnik und der Datenbanktechnik. „Modelle sind die Artefakte der Software-Entwicklung“ [7]. Dementsprechend hoch ist der Stellenwert der Modellierung in der Softwaretechnik. Als einfacher Indikator möge die Tatsache dienen, dass die Termini „model“ und „modeling“ im „Software Engineering Body of Knowledge“ (SWEBOK) [1] fast dreihundert Mal vorkommen.

Selbst wer „Model-driven xyz“ bzw. „model-based xyz“ mit $xyz = \{\text{requirements, architecture, development, testing, ...}\}$ als Modeerscheinungen abtut, kann daher nicht bestreiten, dass Modellierung Bestandteil der Ausbildung in Informatik sein muss. Die Frage, warum Modellierung im Informatikstudium gelehrt werden soll, dürfte damit hinreichend geklärt sein. Wir formulieren dies als erste These:

These 1. Modellierung soll ein expliziter Bestandteil jedes Informatikstudiengangs sein.

Abstract

Modeling has always been an important topic in Informatics education, not only since the advent of “model-driven xx” has made it a mainstream topic. This article presents and discusses twelve theses about the Why, What, Where, When, How, and How Much of modeling in Informatics curricula. The theses are based on the author’s experience from having taught a modeling course at the University of Zurich since more than ten years and discussions with colleagues in the framework of the GI workshop series on modeling. Additionally, a survey conducted among the Informatics graduates at the University of Zurich yields empirical evidence for several of the theses.

Was sollen wir lehren?

Die nachstehenden neun Thesen formulieren die zentralen Lernziele. Sie gelten typischerweise für Bachelorstudiengänge sowie für Diplomstudiengänge (soweit es diese noch gibt). Daran anschließend werden die Thesen kurz diskutiert und begründet.

These 2. Die Studierenden sollen die grundlegenden, gebietsübergreifenden Phänomene der Modellierung lernen (Theorie, Grundsätze der Modellbildung, Definition und Manipulation von Modellen, Metamodelle).

These 3. Die Studierenden sollen die Konzepte der Modellierung von Strukturen, Abläufen, Verhalten und Interaktion in Informatikproblemen verstehen und anwenden.

These 4. Die Studierenden sollen lernen, welches die typischen Artefakte sind, die aus Modellen bestehen oder in denen Modelle eine zentrale Rolle spielen, sowie wann und wo diese Artefakte typisch entstehen und verwendet werden.

These 5. Die Studierenden sollen in allen Teildisziplinen der Informatik, mit denen sie im Studium in Kontakt kommen, ihr Wissen über Modellierung auf Probleme der jeweiligen Teildisziplin anwenden.

These 6. Die Studierenden sollen sowohl das Lesen und Verstehen bestehender Modelle als auch die Bildung von Modellen lernen.

These 7. Die Studierenden sollen die Möglichkeiten und Grenzen modellbasierter Verfahren kennen lernen.

These 8. Die Studierenden sollen geeignete Modellierungssprachen in dem Umfang erlernen, der sie zur Anwendung des Konzeptwissens über Modellierung befähigt.

Korollar. Die Lehre von Modellierung ist etwas grundsätzlich Anderes als die Vermittlung von UML-Kenntnissen.

These 9. Das Erlernen des Umgangs mit Modellierungswerkzeugen ist kein notwendiger Bestandteil der Lehre in Modellierung. Dies gilt primär für die universitäre Lehre, wo die Entwicklung konzeptioneller Fähigkeiten im Vordergrund steht. An Fachhochschulen, wo auch berufspraktisches Wissen zu vermitteln ist, wird das Curriculum in der Regel auch den Umgang mit einschlägigen Werkzeugen umfassen.

These 10. Als Voraussetzung für das Lernen von Modellierung benötigen die Studierenden ein Basiswissen in theoretischer Informatik, beispielsweise über Graphen, Automaten und Sprachen.

Angesichts der Bedeutung der Modellierung in der Informatik genügt es nicht, Modelle in naiver und unreflektierter Weise in den entsprechenden Anwendungsgebieten (zum Beispiel in der Softwaretechnik) zu nutzen. Stattdessen braucht es einen Ansatz, welcher der grundsätzlichen, gebietsübergreifenden Bedeutung von Modellen gerecht wird (These 2). Innerhalb der Modellierung gibt es Felder mit deutlich unterschiedlichen Grundkonzepten, dies betrifft insbesondere die Modellierung von Strukturen, Abläufen, Verhalten und Interaktion. Diese Konzepte gilt es zu vermitteln (These 3). Es wäre jedoch falsch, Modellierung nur grundsätzlich und kontextfrei zu vermitteln. Die Studierenden müssen lernen, in welchen Anwendungskontexten Modelle erstellt und verwendet werden und welches die entsprechenden Artefakte sind (These 4). Deshalb darf die Modellierungsausbildung nicht

ausschließlich Sache einer Spezialveranstaltung sein. Vielmehr muss das Studium die Studierenden dazu anhalten, in allen Teildisziplinen der Informatik, in denen Modelle eine Rolle spielen, ihr Modellierungswissen anzuwenden und zu vertiefen (These 5).

Absolventinnen und Absolventen von Informatikstudiengängen brauchen einerseits die Fähigkeit, bestehende Modelle zu lesen, zu interpretieren und zu analysieren. Andererseits müssen sie aber auch in der Lage sein, Probleme zu analysieren und zu lösen, indem sie geeignete Modelle erstellen (These 6). Sie müssen hierzu die Möglichkeiten, aber auch die Grenzen modellbasierter Verfahren kennen lernen (These 7).

Modellierungssprachen sind die konkreten Träger für Modellierungskonzepte. Sie sind von Natur aus weniger fundamental und weniger langlebig als die zu Grunde liegenden Konzepte. In einer Hochschulausbildung, wo die Vermittlung langlebigen Konzeptwissens im Vordergrund steht, hat die Bedeutung der Modellierungssprachen daher hinter diejenige der Konzepte zu treten. Andererseits sind Konzepte ohne Sprachen zur Formulierung von Modellen nicht vernünftig lernbar, weil die Anwendung der Konzepte auf konkrete Probleme nicht gelehrt und geübt werden kann. Das Erlernen geeigneter Modellierungssprachen ist daher notwendig. Hingegen wäre es falsch, nur Modellierungssprachen zu lehren, beispielsweise in reinen UML-Kursen (These 8 und Korollar dazu).

Werkzeuge für die Bearbeitung von Modellen sind Hilfsmittel und daher von untergeordneter konzeptioneller Bedeutung. Werkzeuge sind daher kein notwendiger Bestandteil der Ausbildung, zumal konkrete Werkzeuge von allen Inhalten, die im Rahmen einer Modellierungslehre vermittelt werden können, die kürzeste Lebensdauer haben (These 9). Aus didaktischer Sicht kann es allerdings geboten sein, Werkzeuge im Unterricht einzuführen und zu verwenden, zumal die Studierenden Wert auf Werkzeuge legen (vgl. [10] und Tabelle 5).

Modelle und Modellierungssprachen basieren zu erheblichen Teilen auf informatisch-mathematischen Grundstrukturen wie Graphen, Relationen, Automaten oder formalen Sprachen. Es ist nützlich, wenn diese bereits zur Verfügung stehen, wenn mit der Lehre von Modellierung begonnen wird. Anderenfalls müsste ein erheblicher Teil einer einführenden Modellierungsveranstaltung

der Vermittlung dieser Voraussetzungen gewidmet werden (These 10).

Wie viel Modellierung brauchen wir wo und wann in der Lehre?

Analysiert man bestehende Informatik-Curricula deutschsprachiger Universitäten und technischer Hochschulen sowie einschlägige curriculare Empfehlungen (zum Beispiel [2, 8]), so stellt man eine große Diversität in der Behandlung von Modellierung fest. Die Spanne reicht von Curricula, in denen Modellierung explizit überhaupt nicht vorkommt, über einige Doppelstunden Modellierung, typisch im Rahmen von Anfängervorlesungen in Informatik oder Programmierung bis zu expliziten Modellierungsvorlesungen im Rahmen von zwei bis vier Semesterwochenstunden.

Diese Divergenz lässt sich zu einem erheblichen Teil auf zwei unterschiedliche Wege im Unterricht von Modellierung zurückführen [11]. Auf dem einen Weg wird Modellierung als Aspekt in allen Lehrveranstaltungen, welche Modelle benötigen oder verwenden, ausgebildet; eine eigenständige Lehre der Modellierung gibt es nicht. Auf dem anderen Weg werden Grundsätze und anwendungsübergreifende Phänomene in einer eigenständigen, in der Regel propädeutischen Vorlesung thematisiert. Lehrveranstaltungen, welche Modelle benötigen oder verwenden, können ein entsprechendes Grundwissen voraussetzen und darauf aufbauend die Modellierungskompetenzen vertiefen.

Der erste Weg hat zweifellos gewichtige Vorteile, indem die Ausbildung dann stattfinden kann, wenn die Inhalte auch gebraucht werden. Die verwendeten Modelle können im jeweiligen Anwendungskontext betrachtet werden; die Studierenden lassen sich dadurch besser motivieren.

Auf der anderen Seite hat dieser Weg auch einen gravierenden Nachteil: Grundsätzliches und Gemeinsames in der Modellierung wird nicht oder eher zufällig diskutiert.

Dort, wo eine explizite Modellierungsveranstaltung angeboten wird, hat sie praktisch immer propädeutischen Charakter und ist daher eine Pflichtveranstaltung im ersten oder zweiten Semester. Der Umfang beträgt meist zwei Semesterwochenstunden Vorlesung plus Übungen; dem Verfasser ist lediglich ein Studiengang (der an der Universität Paderborn) bekannt, welcher eine vierstündige Vorlesung über Modellierung vor-

sieht. Dort ist allerdings ein erheblicher Teil der Inhalte, die sich an [14] orientieren, theoretisches Grundwissen (vgl. These 12 unten).

Aus den Erfahrungen des Verfassers mit beiden Wegen in der Lehre resultieren die nachstehenden zwei Thesen zum Wann, Wo und Wie viel:

These 11. Zur Erreichung der grundlegenden Lernziele in Modellierung (vgl. Thesen 2 und 3 sowie 6 und 7) ist eine eigenständige Modellierungslehrveranstaltung besser geeignet als die „en passant“-Ausbildung in Veranstaltungen, welche Modelle benötigen oder verwenden.

These 12. Eine solche grundlegende Modellierungsvorlesung muss einerseits früh im Curriculum platziert werden, damit die einführenden Vorlesungen über Softwaretechnik, Datenbanken, etc. darauf aufbauen können. Auf der anderen Seite muss ein gewisses theoretisches Grundwissen (zum Beispiel über Graphen, Sprachen oder Automaten) bereits vorhanden sein; es sei denn, die Vermittlung dieses Wissens wird in die Modellierungsvorlesung integriert.

Als Konsequenz bietet sich die Positionierung einer Modellierungsvorlesung im zweiten Semester des Bachelorstudiums an.

Die Vermittlung von Modellierungswissen in einer eigenständigen Lehrveranstaltung darf nicht dazu führen, dass das Thema Modellierung an diese Vorlesung delegiert und im weiteren Verlauf des Studiums nicht mehr explizit thematisiert wird. Dies gilt insbesondere, weil die in den Thesen 4 und 5 geforderte Kontextualisierung in einer reinen Modellierungsveranstaltung nur beschränkt geleistet werden kann. Die Lehrveranstaltungen, in denen Modelle benötigt bzw. verwendet werden, müssen das fach- und anwendungsspezifische Modellieren vermitteln und dafür sorgen, dass das Grundwissen aus der Einführungsvorlesung nutzbringend eingesetzt wird.

Empfehlung. In jedem Informatik-Curriculum sollte es eine eigenständige propädeutische Modellierungsvorlesung geben. Gleichzeitig muss Modellierung in allen einschlägigen Fachveranstaltungen (insbesondere in der Software- und Datenbanktechnik) explizit thematisiert und kontextualisiert werden.

Wie soll Modellierung gelehrt werden?

Sucht man nach Arbeiten zur Didaktik der Modellierung auf der Hochschulstufe, so stößt man im Wesentlichen auf Erfahrungsberichte. Einschlägige Quellen sind insbesondere die Tagungsbände der Tagungsreihen „Modellierung“¹, „Modellierung in Lehre und Weiterbildung“², MoDELS Educators Symposium³ und „Software Engineering im Unterricht der Hochschulen (SEUH)“⁴. Über Erfahrungsberichte hinausgehende Arbeiten sind sehr dünn gesät, beispielsweise das Papier von Borner, Paech und Rückert über Modellverstehen vs. Modellerstellung [3] oder Desels Vorschlag, Modellierung über die Synthese formalisierter Ablaufbeschreibungen zu lernen [4].

Dabei gäbe es durchaus eine Reihe grundlegender didaktischer Fragen zu untersuchen, beispielsweise:

- Welches sind die geeigneten Lehrformen (Vorlesungen, Übungen, Praktika)?
- Wo und wie lassen sich neue, webbasierte Lernformen einsetzen?
- Soll in der verfügbaren Zeit Weniges tief oder Vieles im Überblick unterrichtet werden?
- Wie motivieren wir die Studierenden, insbesondere in einer Modellierungsvorlesung im ersten oder zweiten Semester?
- Gehen wir beim Unterricht einer Modellierungstechnik von der sprachlichen Darstellung aus (Primat des Lesens und Verstehens) oder von der Methodik der Modellerstellung (Primat des Schreibens)?
- Welche Rolle spielen Werkzeuge in der Modellierungsausbildung?
- Wie üben und prüfen wir Modellierungskompetenzen?

Ein Beispiel: Modellierung im BSc Informatik an der Universität Zürich

Im Studiengang „Informatik Richtung Wirtschaftsinformatik“ an der Universität Zürich folgen wir

¹ Auf den Webseiten des GI-Querschnittsfachausschusses Modellierung (<http://www.gi-modellierung.de/>) findet sich unter dem Menüpunkt „Bibliografie“ eine Übersicht über die seit 1998 erschienenen Tagungsbände.

² Workshops auf der Modellierung 2006 und 2008 [5, 6].

³ Das MoDELS Educators Symposium ist ein seit 2005 bestehender Workshop auf der internationalen MoDELS Konferenz (<http://www.modelsconference.org>). Die Tagungsbände sind teilweise von den jeweiligen Symposium-Webseiten herunterladbar.

⁴ Die SEUH-Tagungsbände von 1992 bis 1999 sind als Berichte des German Chapter of the ACM im Teubner Verlag erschienen. Die Tagungsbände von 2001 bis 2007 sind im dPunkt Verlag erschienen. Die SEUH findet alle zwei Jahre statt.

der aus unseren Thesen resultierenden Empfehlung und betreiben seit 1996 eine eigenständige, frühe Modellierungsausbildung im Rahmen der Grundvorlesung Informatik II (vgl. These 11). Bis 2004 waren dies zehn Doppelstunden Vorlesung mit zwei freiwilligen Übungen. Mit der Einführung des Bachelorstudiums wurde die Modellierung zu einem eigenständigen Teilmodul im Modul Informatik II mit zwei Semesterwochenstunden Vorlesung sowie Übungen und Zwischentests, die Bestandteil des Leistungsnachweises sind. Das ganze Teilmodul gibt drei Punkte ECTS. Die zeitliche Positionierung im zweiten Semester leitet sich aus These 12 ab.

Die Vorlesung und die zugehörigen Übungen decken insbesondere die in den Thesen 2 und 3 formulierten Ziele ab. Sie legen ferner die Grundlagen bezüglich der Thesen 6 und 7. Bezüglich Sprachen und Werkzeugen folgt die Vorlesung den Postulaten der Thesen 8 und 9. These 10 ist teilweise umgesetzt: Graphen und Relationen beispielsweise sind Stoff der Vorlesung Formale Grundlagen der Informatik I im ersten Semester, während endliche Automaten und Petrinetze in der Modellierungsvorlesung eingeführt werden.

Eine Kontextualisierung (vgl. Thesen 5 und 6) ist im zweiten Semester erst ansatzweise möglich.



Tabelle 1

Aufbau und Inhalt der Vorlesung Modellierung an der Universität Zürich

1. Einführung in die Modellierung

Definitionen und Begriffe, wozu Modelle, das Prinzip der Modellbildung, Modelle in der Informatik

2. Modelltheorie

Charakteristische Merkmale, Sprache und Modell, Operationen auf Modellen, deskriptive und präskriptive Modellbildung, philosophisch-ethische Aspekte von Modellierung und Erkenntnis

3. Datenmodelle

Statische Strukturen und Zusammenhänge: Motivation und Einsatzgebiete, Entity-Relationship-Konzept, Notationsformen, Methodik der Datenmodellierung

4. UML

Geschichte, Konzepte und heutige Rolle der UML

5. Funktionsmodellierung I: Steuerflussmodelle

Ablaufstrukturen (u.a. Nassi-Shneiderman und Jackson), Aufrufstrukturen, Entscheidungstabellen

6. Funktionsmodellierung II: Datenflussmodelle

Das Datenflussprinzip, Datenflussdiagramme, Strukturierte Analyse, andere Verfahren

7. Verhaltensmodelle

Grundlagen der Verhaltensmodellierung, Zustandsautomaten, Statecharts, Petrinetze

8. Klassen- und Objektmodelle

Motivation und Grundlagen, Objekte und Klassen, Intension und Extension, Klassenmodelle (Prinzipien, Notation, Analysemethoden), Objektmodelle

9. Modellierung von Arbeitsprozessen

Grundlagen und Begriffe, Ereignisgesteuerte Prozessketten (EPK), andere Sprachen

10. Interaktionsmodelle

Interaktion zwischen Menschen und Informatiksystemen: Motivation und Einsatzgebiete, Modellierung des Systemkontextes, Anwendungsfälle und Szenarien

11. Systemmetaphern

Leitbilder für Systemstrukturen und den Umgang mit Systemen: Rolle von Metaphern, ausgewählte Präsentations-, Struktur- und Architekturmetaphern

12. Abstraktionen

Das Prinzip der Abstraktion, Rolle der Abstraktion für das Verstehen von Modellen, Abstraktionsformen: Klassifikation, Komposition, Generalisierung, Benutzung

13. Metamodelle

Grundlagen und Motivation, linguistische und ontologische Metamodelle, höhere Metamodelle

Sie erfolgt schwerpunktmäßig in den Pflichtvorlesungen Software Engineering (drittes Semester) und Datenbanksysteme (viertes Semester), aber beispielsweise auch in den Vorlesungen über Wirtschaftsinformatik, in denen vor allem die Prozessmodellierung wieder aufgegriffen und vertieft wird, oder in den Vorlesungen über Kommunikation und verteilte Systeme.

Die damalige Entscheidung zur Einführung einer eigenständigen Modellierungsausbildung basierte auf Beobachtungen über unzureichende Modellierungsfertigkeiten der Studierenden im damaligen Studiengang mit „integrierter“ Modellierungsausbildung in Vorlesungen über Programmierung, Datenbanken und angewandte Informatik. Wir stellten beispielsweise fest, dass viele Studierende grundsätzliche Modellierungstechniken nicht beherrschten, zum Teil völlig naive Vorstellungen von „der Abbildung der Realität“ in Informatiksysteme hatten und sich oft gar nicht bewusst waren, dass und wie sie Modelle verwendeten [11].

Tabelle 1 gibt einen Überblick über die Inhalte der Vorlesung. Wir beginnen und enden mit Theorie. Dazwischen geben wir eine Einführung in die gängigsten Techniken zur Modellierung von Daten, Abläufen, Flüssen, Verhalten, Objekten, Prozessen und Interaktionen. Das didaktische Prinzip dabei ist, jeweils zunächst die Konzepte anhand einer typischen Sprache einzuführen und daran anschließend die zugehörige Methodik der Modellbildung zu betrachten. Dort, wo es sich anbietet, verwenden wir UML als Sprache. Die Vorlesung ist aber bewusst nicht als UML-Einführung konzipiert, weil das Gewicht auf den Konzepten und nicht auf den Details einer Modellierungssprache liegt. Das Kapitel über UML ist daher mit Absicht sehr kurz gehalten.

Die Vorlesung wird von einem ausführlichen Folienskript begleitet, welches frei zugänglich ist [12].

Während sich das von der Sache her gut bewährt, kämpfen wir mit dem Grundproblem jeder propädeutischen Ausbildung: Wir geben in dieser Vorlesung vielfach Antworten auf Fragen, welche die Studierenden bisher nie gestellt haben, weil sie mit den Problemen, die durch Modellierung gelöst werden, bisher nicht oder nur ansatzweise konfrontiert worden sind.

Empirische Resultate

In diesem Kapitel stellen wir den weiter oben aufgestellten Thesen empirische Evidenz aus einer

Befragung von Informatikabsolventen der Universität Zürich gegenüber. Der Fragebogen wurde an die 570 Mitglieder unserer Alumniorganisation (Alumni Wirtschaftsinformatik Universität Zürich) verteilt. 90 haben geantwortet, die Rücklaufquote beträgt also 16%. Etwa die Hälfte aller Absolventinnen und Absolventen sind Mitglieder der Alumni. Die eingegangenen Antworten können daher als einigermaßen repräsentativ für die Gesamtheit der Absolventinnen und Absolventen betrachtet werden, mit dem üblichen Vorbehalt, dass bei solchen Fragebogenaktionen die engagierten und initiativen Personen stets überrepräsentiert sind. 54 Antworten stammen von Absolventinnen und Absolventen, die im zweiten Semester Modellierung als Pflichtvorlesung besucht haben.

Die als erstes gestellte Frage, ob Modellierung ein expliziter Bestandteil des Studiums der Informatik/Wirtschaftsinformatik sein sollte, findet sehr hohe Zustimmung (Tabelle 2). Diese wird damit empirisch gestützt.

Die praktische Bedeutung der Modellierungsausbildung im Informatikstudium spiegelt sich in den Antworten auf die Frage, wie oft die befragte Person ihr Wissen über Modellierung aus dem Studium in ihrem Berufsleben braucht: fast 70% der Antwortenden brauchen es häufig oder regelmäßig (Tabelle 3).

Die Unterschiede in den Antworten zwischen denjenigen Befragten, welche im zweiten Semester Modellierung als Pflichtvorlesung besucht haben und den übrigen Befragten sind nicht so groß, dass sich eine separate Auswertung im Rahmen dieses Artikels aufdrängt. Auffällig und damit erwähnenswert ist, dass die erste Gruppe die Notwendigkeit einer expliziten Behandlung von Modellierung im

Sollte Modellierung ein expliziter Bestandteil des Studiums der Informatik/Wirtschaftsinformatik sein?

Ja, unbedingt	71	81%
Eher ja	15	17%
Eher nein	2	2%
Nein, nicht nötig	0	0%

Tabelle 2

Brauchen Sie das im Studium erworbene Wissen über Modellierung in Ihrem Berufsleben?

Regelmäßig	23	26%
Häufig	38	43%
Selten	22	25%
Nie	5	6%

Tabelle 3

Wie sollte die Gewichtung zwischen grundlegenden Modellierungskonzepten und konkreten Modellierungssprachen sein?

Nur Konzepte	0	0%
Konzepte im Vordergrund, Sprachen ergänzend	65	73%
Sprachen im Vordergrund, Konzepte ergänzend	24	27%
Nur Sprachen	0	0%

Tabelle 4

von Verhalten erklärt sich aus der Ausrichtung des Informatikstudiums an der Universität Zürich (Informatik Richtung Wirtschaftsinformatik), was dazu führt, dass die weitaus meisten unserer Absolventinnen und Absolventen im Umfeld von betrieblichen Informatikanwendungen tätig sind, wo Verhaltensmodellierung eine deutlich geringere Rolle spielt als in technischen Anwendungen. Auffällig ist ferner die tiefe Bewertung des Themas Metamodellierung. Dies mag damit zusammenhängen, dass Metamodellierung erst in den letzten Jahren ins breite Bewusstsein gerückt ist, während es davor ein exotisches Thema war, mit dem sich nur wenige Spezialisten auskannten. Hervorzuheben ist ferner die starke Bewertung von Grundlagen und Theorie, was These 2 unterstützt.

Zusätzlich zu den abgefragten sechs Rubriken konnten die Befragten weitere aus ihrer Sicht wichtige Inhalte aufführen. Hier wurden vor allem Themen genannt, welche die in den Thesen 4 und 5 genannte Kontextualisierung betreffen, zum Beispiel Einbettung und Methodik der Model-

Studium stärker bejaht (87% gegen 71% „Ja, unbedingt“, vgl. Tabelle 2) und ihr Modellierungswissen im Berufsleben häufiger braucht (74% gegen 60% regelmäßig oder häufig, vgl. Tabelle 3).

Abbildung 1 zeigt die Ergebnisse der Frage nach den wichtigen Lerninhalten. Die sechs genannten Rubriken waren vorgegeben und konnten mit den im Bild genannten vier Stufen bewertet werden. Zusätzlich konnten weitere Themen genannt werden.

Die Resultate stützen These 3. Das vergleichsweise schlechte Abschneiden der Modellierung

Sollte die Benutzung von Modellierungswerkzeugen explizit im Studium gelehrt und geübt werden?

Ja, unbedingt	15	17%
Eher ja	36	40%
Eher nein	30	34%
Nein, nicht nötig	8	9%

Tabelle 5

Grundlagen der Modellbildung, Modelltheorie
Modellierung von Strukturen (z.B. Klassen)
Modellierung von Verhalten (z.B. Zustandsautomaten)
Modellierung von Abläufen (z.B. Aktivitäten, Geschäftsprozesse)
Modellierung von Interaktion (z.B. Anwendungsfälle/Use Cases)
Metamodelle

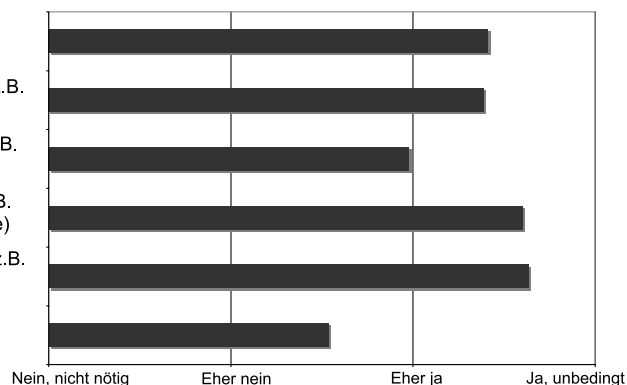


Abb. 1
Modellierungsthemen
und ihre Bewertung



Wo und wann sollte Modellierung gelehrt werden?

Nur in einer (obligatorischen) Einführungsvorlesung	2	2%
Vermittlung der Grundlagen in einer Einführungsvorlesung, Vertiefung und Anwendung in Fachvorlesungen über Datenbanken, Software Engineering, etc.	72	80%
Hauptsächlich als Bestandteil einschlägiger Fachvorlesungen über Datenbanken, Software Engineering, etc.; dazu eine optionale Vertiefungsvorlesung in Modellierung	14	16%
Nur als Bestandteil einschlägiger Fachvorlesungen über Datenbanken, Software Engineering, etc.	2	2%

lierung in Geschäftsanalyse und Softwareentwicklung, Beziehungen und Abhängigkeiten zwischen verschiedenen Modellarten in Softwareentwicklungsprojekten oder mehr Beispiele aus der Praxis.

Bezüglich des Stellenwerts von Modellierungssprachen und -werkzeugen in der Ausbildung ergibt sich folgendes Bild (Tabellen 4 und 5). These 8 über das Verhältnis von Konzepten und Sprachen wird unterstützt. These 9 hingegen über die Bedeutung von Werkzeugen ist aus Sicht der antwortenden Absolventinnen und Absolventen zu akademisch: eine deutliche Mehrheit wünscht explizites Lehren und Üben der Benutzung von Modellierungswerkzeugen.

Schließlich stellt sich noch die Frage nach der Veranstaltungsform und ihrer Positionierung im Curriculum. Hier wird These 11 deutlich bestätigt (Tabelle 6).

Fazit

In diesem Artikel haben wir anhand von zwölf Thesen die Ausbildung in Modellierung im Rahmen von Informatikstudiengängen an Hochschulen umrissen sowie als Beispiel das Modellierungs-Curriculum an der Universität Zürich dargestellt. Die aus einer Befragung von Absolventinnen und Absolventen gewonnene empirische Evidenz unterstreicht die Notwendigkeit einer expliziten Behandlung des Themas Modellierung im Informatikstudium und bestätigt weitestgehend den eingeschlagenen Weg.

Offen bleiben eine Reihe didaktischer Fragen, insbesondere das Motivationsproblem: Wie können wir Studierende in den ersten Semestern ihres Studiums motivieren oder gar dafür begeistern, sich mit Modellen zu beschäftigen? Letztlich geht es hier um das Grundproblem aller propädeutischen Lehrveranstaltungen: Wie gehen wir mit der Situation um, dass wir notwendigerweise Antworten geben auf Fragen, welche die Studierenden noch nie gestellt haben? Eng damit

verbunden ist auch das Problem der Kontextualisierung: Wie vermitteln wir die notwendigen Anwendungskontexte für Modelle, insbesondere dann, wenn den Studierenden das entsprechende anwendungsfachliche Wissen noch fehlt? Schließlich stellt sich wie überall in der Informatik die Frage nach der Verbindung zwischen Theorie und Praxis. Es ist zu hoffen, dass sich vermehrt Lehrende für diese Fragen interessieren – die für die Verbreitung und den Austausch von Erfahrungen notwendigen Foren, im deutschsprachigen Raum insbesondere die Tagungsreihe „Modellierung“, sind vorhanden und der Nutzen solcher Arbeiten ist evident.

Literatur

1. Abran A, Moore JW (Hrsg) (2004) Guide to the software engineering body of knowledge (SWEBOOK), 2004 Version. IEEE Computer Society Press, Los Alamitos, Ca
2. ACM/IEEE CS Joint Task Force on Computing Curricula (2001) Computing Curricula 2001: Computer Science, Final Report. http://www.acm.org/education/education/education/curric_vols/cc2001.pdf (letzter Zugriff 11.7.2008)
3. Borner L, Paech B, Rückert J (2006) Vom Modellverstehen zum Modellstellen. In: Desel J, Glinz M (Hrsg) (2006) Modellierung in Lehre und Weiterbildung. Workshop auf der Modellierung 2006, 23.3.2006, Innsbruck. Technischer Bericht ifi-2006.03, Institut für Informatik, Universität Zürich, S 7–15
4. Desel J (2008) Modellieren lernen über Formalisieren von Ablaufbeschreibungen. In: Desel J, Glinz M (Hrsg) (2008) Modellierung in Lehre und Weiterbildung. Workshop auf der Modellierung 2008, 13.3.2008, Berlin. Technischer Bericht ifi-2008.04, Institut für Informatik, Universität Zürich, S 37–46
5. Desel J, Glinz M (Hrsg) (2006) Modellierung in Lehre und Weiterbildung. Workshop auf der Modellierung 2006, Innsbruck. Technischer Bericht ifi-2006.03, Institut für Informatik, Universität Zürich
6. Desel J, Glinz M (Hrsg) (2008) Modellierung in Lehre und Weiterbildung. Workshop auf der Modellierung 2008, Berlin. Technischer Bericht ifi-2008.04, Institut für Informatik, Universität Zürich
7. Ebert J (2005) Zitat aus der Podiumsdiskussion „Software-Entwicklung und Modellierung“ auf dem Workshop Modellierung 2005. In: Paech B, Desel J (Hrsg) Workshop Modellierung 2005. <http://www.gi-modellierung.de> "Archiv" Tagungsunterlagen (letzter Zugriff: 11.7.2008)
8. Fakultätentag Informatik (Hrsg) (2004) Empfehlungen zur Einrichtung von konsekutiven Bachelor- und Masterstudiengängen in Informatik an Universitäten. <http://www.ft-informatik.de> (letzter Zugriff: 11.7.2008)
9. Gesellschaft für Informatik e.V. (GI) (Hrsg) (2004) Empfehlungen für Bachelor- und Masterprogramme im Studienfach Informatik an Hochschulen. GI-Empfehlungen, Bonn
10. Glinz M (1996) The teacher: "Concepts!" The student: "Tools!" – On the number and importance of concepts, methods, and tools to be taught in software engineering education, proceedings third international workshop on software engineering education, Berlin. Softwaretech-Trends 16(1):32–34, 55

11. Glinz M (1999) Modellierung im Informatikstudium. In: Desel J, Pohl K, Schürr A (Hrsg) Modellierung '99, Workshop der GI. Teubner, Karlsruhe Stuttgart, S 190–195
12. Glinz M (2008) Skript zur Vorlesung Informatik IIa: Modellierung, Sommersemester 2008. Institut für Informatik, Universität Zürich. http://www.ifi.uzh.ch/rerg/courses/fs08/inf_ii (letzter Zugriff: 11.7.2008)
13. Ludewig J (2003) Models in software engineering – An introduction. Softw Syst Model 2(1):5–14
14. Kastens U, Kleine Büning H (2005) Modellierung: Grundlagen und formale Methoden. Hanser, München
15. Stachowiak H (1973) Allgemeine Modelltheorie. Springer, Wien



Martin Glinz ist ordentlicher Professor für Informatik und Leiter der Forschungsgruppe Requirements Engineering an der Universität Zürich. Er beschäftigt sich hauptsächlich mit Methoden, Sprachen und Werkzeugen zur Modellierung von Anforderungen. Weitere Interessensgebiete

sind Software Engineering, Software-Qualität und Modellierung im Allgemeinen.

Er hat an der RWTH Aachen in Informatik promoviert. Vor seiner Berufung an die Universität Zürich war er zehn Jahre in der Industrie als Forscher, Entwickler, Berater und Dozent im Bereich Software Engineering tätig.